Homework 3 prepared for Prof. M. Khalid Jawed teaching MAE 263F in Fall 2025

Jens Dekker

I. Code description

The code follows the same structure of code implemented for Homework 2 with the core difference being the iteration to find the next best fixed degree of freedom.

Using the initSpringNetwork file, the structure of all the nodes, stretching springs and bending springs was stored into the files in the springNetworkData directory. From there the Homework3 file imports the spring network data. It generates the rest length and Voronoi lengths of the stretch and bending springs along with some other initialization such as time steps, max time, and setting up containers for storing plotting data.

Once everything is initialized, the code begins to iterate over each time step. For each time step a target_x and target_y is defined for where the middle node should be placed as well as the distance between the current position of the middle node and the target location. From here, the code enters a loop while the distance to the target is too great, and the max number of iterations has not been reached. From here the solver calculates the position of the beam given the fixed DOFs. It then recalculates the distance to the target position based on the new position and if the distance is greater than allowed, it will update the position of the last two nodes of the old position to accommodate.

The solver will recalculate the new positions again based on the updated old positions. This process will loop until the distance to the target is small enough, where it moves to the next time step and repeats. Note that the old velocities are not updated during this iteration to preserve the previous state of the beam. This aims to treat the control of the robot's control as an instantaneous adjustment of last two node's position, to which the beam must respond.

By inspection the solution which requires the least energy can be seen as letting the beam swing around the pivot joint at the first node of the beam. To take advantage of this the control path sets its theta value to keep the first, and last two nodes collinear; primarily controlling the beam by the x and y position of the last node. Alternative solutions will be discussed in a later section.

II. Results

A. Control Paths

The simulation created the control paths as shown in Figures 1 - 4.

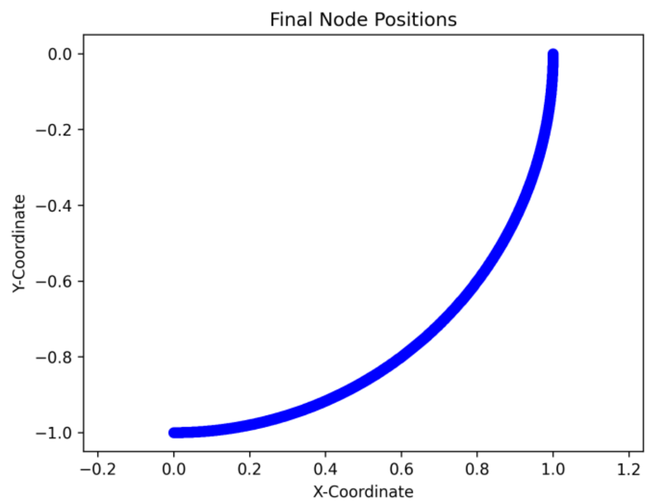

Figure 1: Positions of the Last Node in the Beam

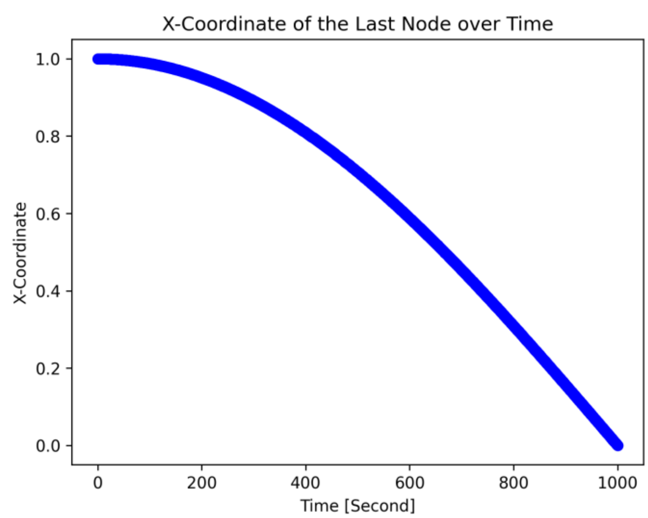

Figure 2: X-Coordinate of the Last Node of the Beam over Time.

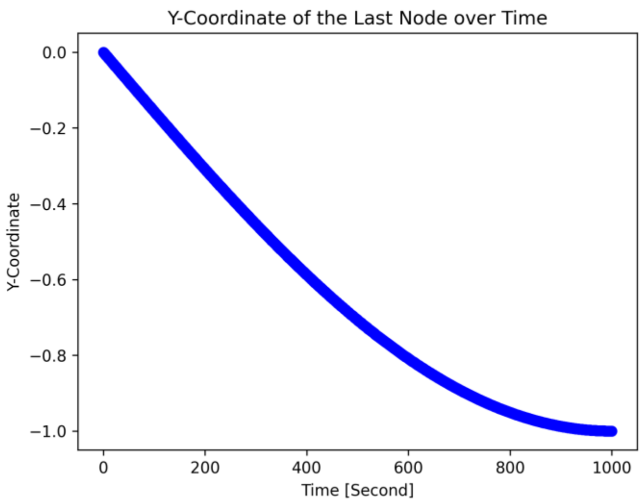

Figure 3: Y-Coordinate of the Last Node of the Beam over Time.

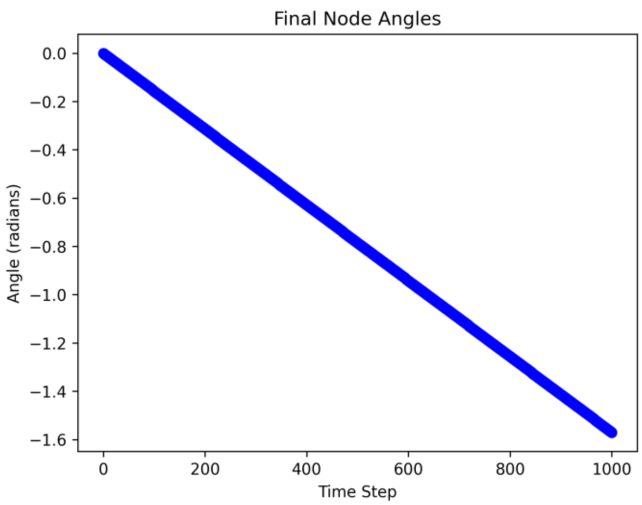

Figure 4: Angle Between the Last Two Nodes of the Beam.

B. Beam Snapshots

Snapshots of the beam in 200 second intervals are in Figures 5-10

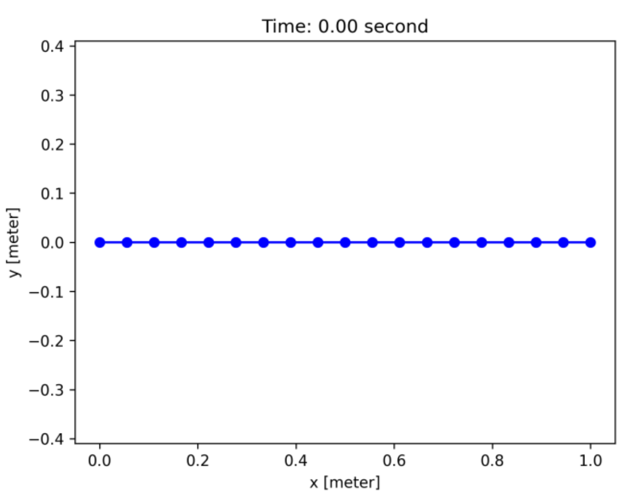

Figure 5: Beam Position at Time = 0 seconds.

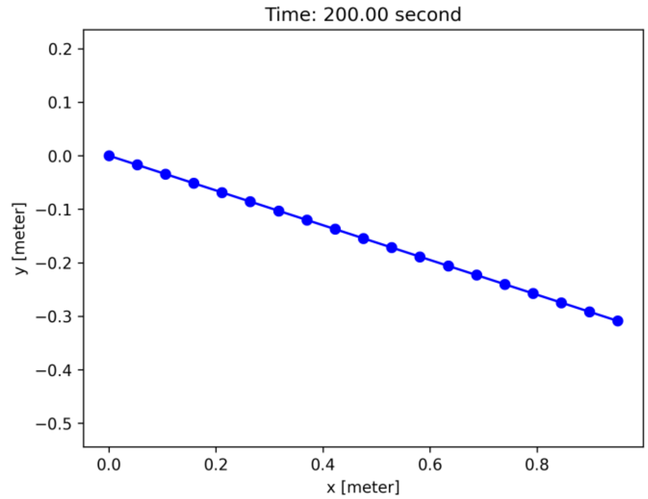

Figure 6: Beam Position at Time = 200 seconds.

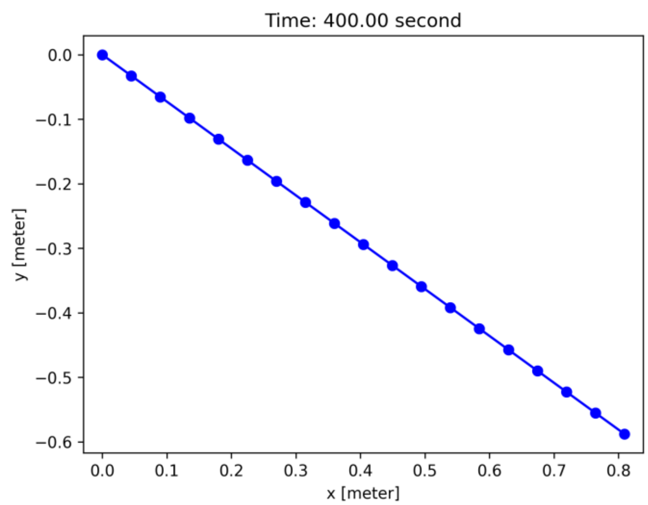

Figure 7: Beam Position at Time = 400 seconds.

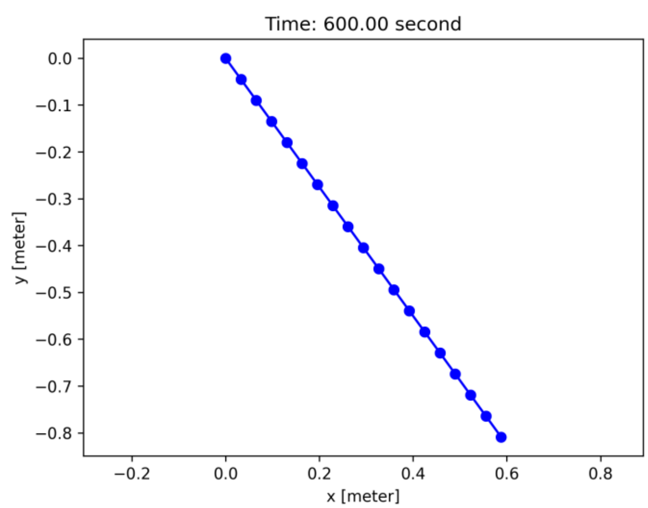

Figure 8: Beam Position at Time = 600 seconds.

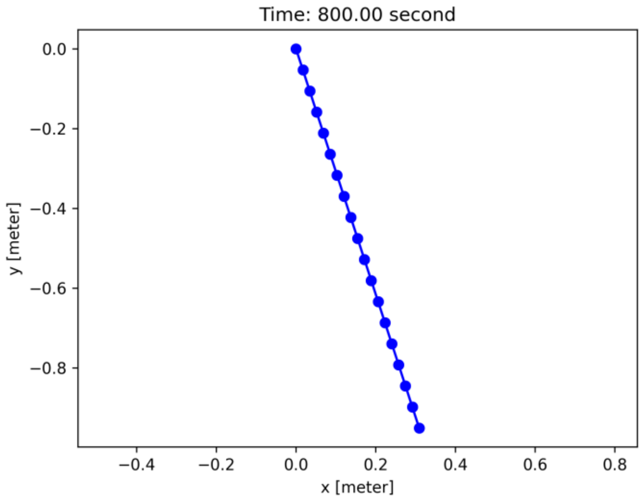

Figure 9: Beam Position at Time = 800 seconds.

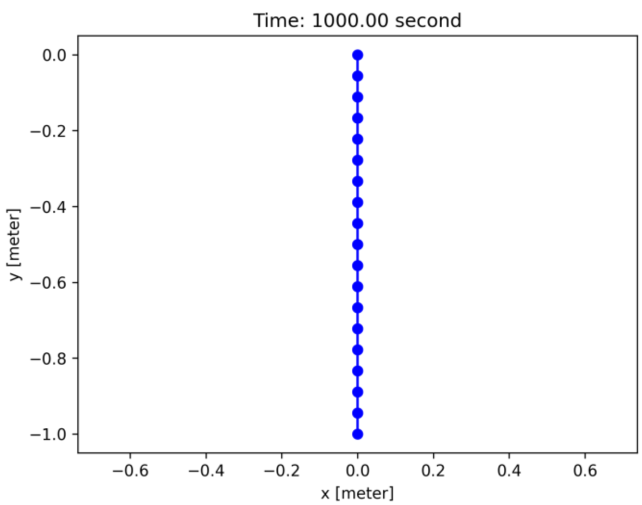

Figure 10: Beam Position at Time = 1000 seconds.

III. Feasibility

The main limitation that this solution faces is the workspace of the robot manipulator. Given the beam length of 1 meter the robot would need to trace a quarter circle with a radius of 1 meter. The implemented solution requires the minimum energy to be input into the beam as there is minimal stretching or bending of the beam.

If another solution is required, the end of the beam could be manipulated in the positive X and negative Y. At time = 500 the beam would be extended, whereas at time = 0 the last two nodes would be pointed downward, and the x and y would be manipulated to achieve the starting position. As the robot moves to achieve the desired path it would turn to align to the origin of the plot and extend to an undeformed length. Then it would follow a symmetric path for the remaining have of the path.

This proposed path would produce a smaller workspace requirement for the robot. However, this would require more torque from the robot to deform the beam in the desired shape. Moreover, additional validation is required to ensure that the beam does move past its elastic limit. The simulated solution does not require energy or elastic limits, but will require a larger workspace.